Red Hat Network 3.6

Channel Management Guide

Red Hat Network 3.6: Channel Management Guide

channel-mgmt(EN)-3.6-RHI (2004-11-09T17:48)

Table of Contents

Chapter 1.
Introduction

This document discusses issues surrounding the deployment and maintenance of customized software channels for RHN Proxy Server and RHN Satellite Server. It should be used after the RHN Satellite Server or RHN Proxy Server is installed and configured. Refer to the *RHN Satellite Server Installation Guide* and the *RHN Proxy Server Installation Guide*, respectively, for details, followed by the *RHN Client Configuration Guide*.

In some instances, this document will refer to actions to be taken on the Red Hat Network Web servers. For Proxy customers, this will refer to the central Red Hat Network Servers at https://rhn.redhat.com. For Satellite customers, this will refer to the RHN Satellite Server at the customer site.

Chapter 2.

Introduction to RHN Channels

A Red Hat Network channel is a collection of packages. Channels help to segregate packages by sensible rules: a channel may contain packages from a specific Red Hat distribution, for instance. A channel may contain packages for an application or family of applications. Users may also define channels for their own particular needs; a company may create a channel that contains packages for all of the organization's laptops, for example.

2.1. Base Channels and Child Channels

There are two types of channels: *base channels* and *child channels*. A base channel consists of packages based on a specific architecture and Red Hat Enterprise Linux release. A child channel is a channel associated with a base channel that contains extra packages.

A system must be subscribed to one base channel and one base channel only. A system can be subscribed to multiple child channels of its base channel. Only packages included in a system's subscribed channels can be installed or updated on that system through Red Hat Network.

When a system is registered with Red Hat Network, it is assigned to the base channel that corresponds to system's version of Red Hat Enterprise Linux. Once a system is registered, its default base channel may be changed to a private base channel; these changes can be made on a per-system basis in the RHN website.

On the Red Hat Network website, the **Channels** page (located under the **Channels** tab in the top navigation bar) provides a list of all base channels and their child channels. Clicking on the name of a channel displays the **Channel Details** page, which provides a list of all of the packages in that channel, its Errata, and any associated systems.

Remember, two different types of ISOs are available through the RHN website. *Distribution ISOs* represent distributions of the Red Hat Enterprise Linux operating system and contain only the packages. *Channel Content ISOs*, visible only to Satellite customers, also contain metadata in the form of XML that enable the Satellite to properly parse and serve the packages. Distribution ISOs can be found in the **Easy ISOs** page and the **Downloads** tab of the respective **Channel Details** page. Channel Content ISOs exist only on the **Downloads** tab of the Satellite's **Channel Details** page.

2.2. Subscribing to Channels

Systems may be subscribed to channels in the following ways:

- Registration through activation keys — Because of the simplicity and speed of activation keys, this is the preferred method for registering systems as clients of either RHN Proxy Server or RHN Satellite Server. Systems registered using an activation key are subscribed to all channels associated with that activation key. For more information on activation keys, consult the *RHN Client Configuration Guide* and the *RHN Management Reference Guide*.
- Install registration — When a system is initially registered through either the **Red Hat Update Agent** or the **Red Hat Network Registration Client**, it is automatically assigned to the base channel that corresponds to the version of Red Hat Enterprise Linux on the system. For more information on using these applications, refer to the respective chapter of the *RHN Reference Guide* for your entitlement level (Management or Provisioning).
- Website subscription — Various specific child channels are available for subscription, depending on the system's base channel. The system may be subscribed to the child channel through the RHN website. Customers who have created their own base channels may also reassign systems to these

custom channels through the website. For more information on subscribing to channels online, refer to the RHN Website chapter of the *RHN Reference Guide*.

2.3. Channel Availability

There are many channels in Red Hat Network. Some are available to all users, some are available to users in a particular organization, and others are available only to customers who have purchased access to them. Channels fall into three main categories:

- Paid Service Channels - These channels are available to customers who have purchased access to them either directly or in conjunction with a particular Red Hat product. Red Hat Enterprise Linux is a paid service channel.
- Custom Channels - These channels are created by customers to manage custom packages. These channels, also known as private channels, appear only to the organization who creates them; they can never be accessed by anyone else.

  For immediate and secure service, a customer may create a custom channel for an organization and host the channel on the organization's private RHN Satellite Server or RHN Proxy Server. This document focuses on that process.

2.4. Tools, Repositories, and Practices

Before creating and managing channels, you should note the differences between the various tools and repositories at your disposal. This is especially important if you will be deploying both a RHN Satellite Server and RHN Proxy Server, as this increases the utilities and storage locations available. Further, a Proxy-Satellite combination offers certain best practices for optimal performance.

First, become familiar with these package management tools:

- **RHN Package Manager** - This application is used to push custom packages into custom channels on an RHN Proxy Server.
- **RHN Push** - This application is used to push custom packages into custom channels on an RHN Satellite Server.
- **RHN Satellite Synchronization Tool** - This application is used to import and synchronize standard packages on an RHN Satellite Server with Red Hat Network. either via the Internet or CD-ROM.

Each of these tools has a corresponding package repository. Both the **RHN Package Manager** and the **RHN Push** require the creation of a temporary staging directory for placement of custom packages that will be uploaded to the Proxy or Satellite. The **RHN Satellite Synchronization Tool** may also necessitate a temporary directory to hold channel ISOs, if the Satellite is not synchronized over the Internet. All three of these repositories should be deleted after use. Red Hat recommends first archiving the custom packages used by **RHN Package Manager** and **RHN Push**.

Customers using both RHN Proxy Server and RHN Satellite Server should be using only the **RHN Push** (as well as the **RHN Satellite Synchronization Tool** for importing and syncing) because this combination requires custom packages and channels be uploaded to the Satellite only. From there, the Proxies will obtain the packages and distribute them to client systems.

Chapter 3.
Building Custom Packages

There are many things that might go wrong when building packages. This is especially true when these packages must be delivered and installed through Red Hat Network. This chapter describes how to build packages of the highest quality that may be successfully delivered via Red Hat Network.

3.1. Building packages for Red Hat Network

Red Hat Network uses the *RPM Package Manager* (RPM) technology to determine what software additions and updates are applicable to each client system. Thus, all packages retrieved from Red Hat Network must be in RPM format. (Entire ISO images, however, are also available through the **Software** tab of the RHN Web interface.)

RPM is an extremely powerful tool that provides users with a simple method for installing, uninstalling, upgrading, and verifying software packages. It also allows software developers to package the source code and compiled versions of a program for end users.

3.1.1. RPM Benefits

RPM provides the following advantages:

Easy Upgrades

Using RPM, you can upgrade individual components of a system without completely reinstalling. When Red Hat releases a new version of Red Hat Enterprise Linux, users do not have to reinstall (as they do with operating systems based on other packaging systems). RPM allows intelligent, fully-automated, in-place upgrades of your system. Configuration files in packages are preserved across upgrades so users do not lose their customizations. There are no special upgrade files needed to update a package because the same RPM file is used to install and upgrade the package.

Powerful Querying

RPM is designed to provide powerful querying options. Users can search through their entire RPM database for all packages or just for certain files. Users can also easily find out what package a file belongs to and from where the package came. The files an RPM package contains are in a compressed archive, with a custom binary header containing useful information about the package and its contents, allowing you to query individual packages quickly and easily.

System Verification

Another powerful feature is the ability to verify packages. If users are worried they deleted an important file from a package, they can simply verify the package. They will be notified of any anomalies. If errors do exist, they can reinstall the questionable package easily. Modified configuration files are preserved during reinstallation.

Pristine Sources

A crucial design goal was to allow the use of *pristine* software sources, as distributed by the original authors of the software. With RPM, the pristine sources can be packaged, along with any patches that were used, plus complete build instructions. This is an important advantage for several reasons. For instance, if a new version of a program is released, users do not necessarily have to start from scratch to make it compile. Users can look at the patch to see what they *might* need to do. All the compiled-in defaults and changes made to get the software to build properly are easily visible using this technique.

Keeping sources pristine may seem important only to developers, but it results in higher quality software for end users, as well.

3.1.2. RHN RPM Guidelines

The strength of RPM lies in its ability to define dependencies and identify conflicts accurately. Red Hat Network relies heavily on this aspect of RPM. Red Hat Network offers an automated environment, which means that no manual intervention can take place during the installation of a package. Therefore, when building RPMs for distribution through Red Hat Network, it is imperative to follow these rules:

1. Learn RPM. It is crucial to have a fundamental understanding of the important features of RPM to build packages properly. For more information about RPM, consult the following resources:

   - http://www.rpm.org/RPM-HOWTO/
   - http://www.redhat.com/docs/books/max-rpm/
   - http://www.rpm.org/mailing_list/

2. When building an RPM for a child channel, build the package on a fresh install of Red Hat Enterprise Linux of the same version as the child's base channel. Be sure to apply all updates from Red Hat Network first.
3. The RPM package must be able to be installed without using the `--force` or `--nodeps` options. If you cannot install an RPM cleanly on your build system, Red Hat Network will not be able to install it automatically on a user's system.
4. The RPM package filename must be in the NVR (name, version, release) format and must contain the architecture for the package. The proper format is name-version-release.arch.rpm. For example, a valid RPM package filename is `pkgname-0.84-1.i386.rpm`, where name is *`pkgname`*, version is *`0.84`*, release is *`1`*, and arch is *`i386`*.
5. The RPM package should be signed by the maintainer of the package. Unsigned packages may be distributed through Red Hat Network, but the **Red Hat Update Agent** (`up2date`) must be forced to accept them. Signing packages is recommended and is covered in Section 3.2 *Digital Signatures for RHN Packages*.
6. If the package is changed in any way, including changing the signature or recompiling, the version or release must be increased incrementally. In other words, the NVRA (including architecture) for each RPM distributed through RHN must correspond to a unique build to avoid ambiguities.
7. No RPM package may obsolete itself. This may seem obvious, but it bears mentioning.
8. If an RPM package is split into separate packages, be extremely careful with the dependencies. Do not split an existing package unless there is a compelling reason to do so.
9. No RPM package may rely upon interactive pre-install, post-install, pre-uninstall or post-uninstall scripts. If the RPM requires direct user intervention during installation, it will not work with Red Hat Network.
10. Any pre-install, post-install, pre-uninstall, and post-uninstall scripts should never write anything to stderr or stdout. Redirect the messages to `/dev/null` if they are not necessary or write them to a file.
11. When creating the spec file, use the group definitions from `/usr/share/doc/rpm-*/GROUPS`. If there is not an exact match, select the next best match.
12. Use the RPM dependency feature to make sure the program will run after it is installed.

13. It may be tempting to create an RPM by archiving files and then unarchiving them in the post-install script, but do not do it. This defeats the purpose of RPM. If the files in the archive are not included in the file list, they cannot be verified or examined for conflicts. In the vast majority of cases, RPM itself can pack and unpack archives most effectively anyway. For instance, don't create files in a %post that you don't clean up in a %postun section.

3.2. Digital Signatures for RHN Packages

All RPM packages distributed through RHN should have a *digital signature*. A digital signature is created with a unique private key and can be verified with the corresponding public key. After creating a package, the SRPM (Source RPM) and the RPM can be digitally signed with a GnuPG key. Before the package is installed, the public key can be used to verify the package was signed by a trusted party and the package has not changed since it was signed.

3.2.1. Generating a GnuPG Keypair

A GnuPG keypair consists of the private and public keys. To generate a keypair, as root at a shell prompt, type the following command:

```
gpg --gen-key
```

If you execute this command as a non-root user, you will see the following message:

```
gpg: Warning: using insecure memory!
```

This message appears because non-root users cannot lock memory pages. If such users could lock memory pages, they could perform out-of-memory denial of service attacks. Since you do not want anyone else to have your private GnuPG key or your passphrase, you should generate the keypair as root. The root user can lock memory pages, which means the information is never written to disk.

After executing the command to generate a keypair, you will see an introductory screen containing key options similar to the following:

```
gpg (GnuPG) 1.0.7; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
   (5) RSA (sign only)
Your selection?
```

Accept the default option: `(1) DSA and ElGamal`. This option will allow you to create a digital signature and encrypt (and decrypt) with two types of technologies. Type **1** and then press [Enter].

Next, choose the key size or how long the key should be. The longer the key, the more resistant against attacks your messages will be. Thus, creating a key of at least 1024 bits in size is recommended.

The next option asks you to specify how long you want your key to be valid. If you do choose an expiration date, remember that anyone with whom you exchanged your public key will also have to be informed of its expiration and supplied with a new public key. It is recommended that you select no expiration date. If you do not choose an expiration date, you will be asked to confirm your decision:

```
Key does not expire at all
```

```
Is this correct (y/n)?
```

Press [y] to confirm your decision.

Your next task is to provide a User ID containing your name, your email address, and an optional comment. Each will be requested individually. When you are finished, you will be presented with a summary of the information you entered.

Once you accept your choices, you will have to enter a passphrase.

Tip

Like your account passwords, a good passphrase is essential for optimal security in GnuPG. Mix your passphrase with uppercase and lowercase letters, use numbers, and/or include punctuation marks.

Once you enter and verify your passphrase, your keys will be generated. You will see a message similar to the following:

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++.+++++.++++++++....++++++++++..+++++.+++++.+++++++.+++++++
+++.++++++++++++++++++++++++++++++++++++++.........................++++
```

When the activity on the screen ceases, your new keys will be made and placed in the directory `.gnupg` in root's home directory (since you are running the command as root). To list your keys, (still as root) use the command:

```
gpg --list-keys
```

You will see something similar to the following:

```
/root/.gnupg/pubring.gpg
------------------------------------
pub  1024D/B7085C8A 2002-02-18 Your Name <you@example.com>
sub  1024g/E12AF9C4 2002-02-18
```

To retrieve your public key, use the following command:

```
gpg --export -a 'Your Name' > public_key.txt
```

Your public key will be written to the file `public_key.txt`.

This public key is quite important. It's the key that must be deployed to all client systems that will receive custom software through `up2date`. Techniques for deploying this key across an organization are covered in the *Red Hat Network Client Configuration Guide*.

3.2.2. Signing packages

Before you can sign packages, you need to configure your `~/.rpmmacros` file to include the following:

```
%_signature gpg
%_gpg_name AB6E2B72
```

The `_gpg_name` is the key ID, which tells **RPM** which signature in your GPG keyring to use. This value can be derived from the command:

```
gpg --list-keys
```

To sign the package *`package-name-1.0-1.noarch.rpm`*, use the following command:

```
rpm --resign package-name-1.0-1.noarch.rpm
```

Enter your passphrase. To make sure the package is signed, use the following command:

```
rpm --checksig -v package-name-1.0-1.noarch.rpm
```

You should see the phrase `Good signature from "`*`Your Name`*`"` in the output.

Chapter 4.

Custom Channel and Package Management

Custom channels allow administrators to use the Red Hat Network infrastructure to deploy packages built and maintained by their organizations. All channel and package management activities will take place in the **Channels** tab of the RHN website. The instructions here should be used in conjunction with the RHN website chapter of the *RHN Reference Guide*.

Because of the potential problems that may arise from deploying untested packages throughout your production environment, Red Hat strongly recommends creating beta channels covering select systems that can be used for staging. For instance, if you have a system group of Web servers that will receive a set of custom packages, we suggest using your channel management capabilities to install the packages on a non-critical subset of representative systems first. These temporary channels can then be deleted using the steps described in Section 4.8 *Deleting Software Channels*.

4.1. Channel Management Privileges

In order to perform any channel management tasks, users must have obtained the proper permissions. These permissions can be modified through the Red Hat Network website. Permissions are assigned to users by Organization Administrators. Channel Administrator privileges are assigned as follows:

1. Log into the Red Hat Network website as an Organization Administrator.
2. On the top navigation bar, click the **Users** tab and then click the name of the user who will be performing channel management functions.
3. On the **User Details** page, scroll down to the **Roles** section and select the checkbox labeled **Channel Administrator**. Then click **Update** at the bottom of the page. Note that Organization Administrators are automatically granted channel administration privileges.
4. Have the user log into the Red Hat Network website, click the **Channels** tab on the top navigation bar, and ensure the **Manage Software Channels** button appears in the corresponding left navigation bar.

4.2. Manage Software Channels

In addition to the buttons and pages available to standard RHN Management-level users, RHN Satellite Server and RHN Proxy Server customers also have access to **Manage Software Channels** in the left navigation bar. This button opens the **Software Channel Management** interface, where all custom software channel management work occurs.

Warning

Customers using both RHN Proxy Server and RHN Satellite Server should be managing custom channels and packages only on the Satellite, since the Proxies should be receiving updates directly from it. Managing packages and channels on a Proxy in this combined configuration risks putting your servers out-of-sync.

Clicking links within the Software Channel Management list takes you to different tabs of the **Managed Software Channel Details** page. Clicking a channel name opens the **Details** tab, while clicking its number of packages opens the **List/Remove** subtab of the **Packages** tab. Refer to Section 4.3 *Managed Software Channel Details* for a full explanation of these areas.

4.3. Managed Software Channel Details

Virtually all custom channel management tasks are carried out within the **Managed Software Channel Details** page, accessed by clicking **Manage Software Channels** in the left navigation bar and then the name of channel to be altered. This page consists of two primary tabs: **Details** and **Packages**.

- **Details** — Provides basic information about the channel, such as its parent channel, name, summary, and description. Some of this information is modifiable. In addition, a Globally Subscribable checkbox can be seen by Organization Administrators and Channel Administrators. This signifies the default behavior of every channel allowing any user to subscribe systems to it. Unchecking this box and clicking **Update Channel** causes the appearance of a **Subscribers** tab, which may then be used to grant certain users subscription permissions to the channel.
- **Subscribers** — Lists users who have subscription permissions to the custom channel. This tab appears on two conditions: First, the logged in user must be an Organization Administrator or a Channel Administrator. Second, the Globally Subscribable checkbox on the **Details** tab must be unchecked, thereby making the channel subscribable by user. On this tab, select the checkboxes of the users to be allowed to subscribe systems to this channel and click **Update**. Note that Organization Administrators and Channel Administrators automatically have subscription access to all channels.
- **Managers** — Lists users who have management permissions to the custom channel. This tab appears for Organization Administrators and Channel Administrators. Select the checkboxes of the users to be allowed full administration of this channel and click **Update**. This status does not enable the user to create new channels. Note that Organization Administrators and Channel Administrators automatically have management access to all channels.
- **Errata** — Provides the Errata associated with each of your custom channels. This tab contains subtabs that allow you to view, add, remove, and clone Errata: **List/Remove**, **Add** and **Clone**. Note that cloning Errata can be done only via RHN Satellite Server.
  - **List/Remove** — Displays all of the Errata currently associated with the custom channel and provides a means to cancel that association. To remove Errata from the channel, select their checkboxes and click **Remove Errata** on the bottom right-hand corner of the page. A confirmation page appears with the Errata to be removed listed. Click **Confirm** to complete the action.
  - **Add** — Enables the addition of Errata to the channel. All of the Errata potentially applicable to the channel are listed. To add Errata to the channel, select the appropriate checkboxes and click **Add Errata**. Refer to Chapter 5 *Custom Errata Management* for a comprehensive discussion of Errata management.
  - **Clone** — Allows Satellite customers to replicate Errata and associated packages for a cloned channel. This subtab immediately appears populated for channels that were cloned with either the original state or select Errata option but will also gain Errata whenever one is issued for the target (or originating) channel. This makes it useful for channels cloned with the current state option, as well. Refer to Section 4.7 *Cloning Software Channels* for a full discussion of cloning options.

    To include Errata from the target channel in the cloned channel, select either Merge or Clone from each advisory's dropdown menu. The Merge option exists only if the Erratum has been previously cloned. Use it to associate the Erratum across channels and avoid duplicate entries. Use the Clone option to create a new entry, such as when modifying it from the previous clone. By default, cloned Errata inherit the label of the original Red Hat advisory (ex. RHSA-2003:324) with the "RH" prefix replaced with "CL". Subsequent clones of the same advisory have their second letters bumped to denote their order, such as "CM" and "CN". These labels can be altered through the **Managed Errata Details** page. Refer to Section 5.2 *Managed Errata Details* for instructions.

    In addition to the Merge option, previously cloned Errata contain values within the **Owned Errata** column. The Erratum label is linked to its details page. The "pub" and "mod" flags within

parentheses identify whether the cloned Erratum has been published or modified from the original advisory. A plus sign (+) before the flag indicates affirmative, the cloned Errata has been published. A minus sign (-) before the flag denotes negative, for instance "-mod" may mean a package has been deleted. To find out more about publishing and editing custom Errata, refer to Section 5.1 *Manage Errata*.

To exclude Errata from the cloned channel, select Do Nothing from their dropdown menus. When satisfied with the changes, click **Clone Errata**. Review the impending changes on the confirmation page and click **Update Errata**.

- **Packages** — Provides the packages associated with each of your custom channels. This tab contains subtabs that allow you to view, add, and remove packages: **List/Remove** and **Add**.
  - **List/Remove** — Displays all of the packages currently associated with the custom channel and provides a means to cancel that association. To remove packages from the channel, select their checkboxes and click **Remove Packages** on the bottom right-hand corner of the page. A confirmation page appears with the packages to be removed listed. Click **Confirm** to complete the action.


Important

Users should note that this list differs from the package list available through the standard **Software Channel Details** page in that it displays all versions of a package remaining in the database, rather than just the latest. Therefore, you may revert to a previous version of a package simply by removing the latest version.

  - **Add** — Enables the addition of packages to the channel. To see available packages, select an option from the **View** dropdown menu and click **View**. To add packages to the channel you are editing, select the appropriate checkboxes and click **Add Packages**. Refer to Section 4.6 *Assigning Packages to Software Channels* for a comprehensive discussion of this process.
  - **Compare** — Enables the comparison of package lists between different channels. To see the differences, select another channel from the **Compare to:** dropdown menu and click **Compare**. A list appears showing all packages not contained by both channels and indicating the existing channel location of each.

4.4. Manage Software Packages

In addition to adding packages to and removing them from channels, RHN Satellite Server and RHN Proxy Server users also have the option of deleting packages entirely from both the database and filesystem (although removal from the filesystem is delayed by about one hour). This can be done through the **Software Package Management** page, accessed by clicking **Manage Software Packages** in the left navigation bar.


Warning

Although deleting packages from the database can be undone by uploading them again, they will lose their association with any Errata. Upon reloading, they will have to be reassociated with Errata manually. Refer to Chapter 5 *Custom Errata Management* for instructions.

To remove packages from the database, in the **Software Package Management** page, select an option containing them from the **View** dropdown menu and click **View**. Then select the appropriate check-

boxes and click **Delete Packages**. A confirmation page will appear with the packages listed. Click **Confirm** to delete the packages entirely.

Remember, since the actual RPMs are stored on the RHN Proxy Server, its custom packages cannot be downloaded through the RHN website, although they are listed. They must be retrieved by the client system using `up2date`. Since the RHN Satellite Server provides its own website, its custom packages are accessible via the Web or the **Red Hat Update Agent**. Regardless, to obtain custom packages, the client system must be subscribed to the channel containing the packages.

4.5. Creating a Software Channel

Before uploading packages to the server, a custom channel can be created to house them. Refer to Chapter 6 *Uploading and Maintaining Custom Packages* for instructions. Once uploaded, packages may be reassigned through the website, as described in Section 4.6 *Assigning Packages to Software Channels*.

Channels are created on the Red Hat Network website as follows:

1. Log into the Red Hat Network website as a Channel Administrator.
2. On the top navigation bar, click the **Channels** tab and then click the **Manage Software Channels** button in the corresponding left navigation bar.
3. On the **Software Channel Management** page, click **create new software channel** at the top-right corner. RHN Satellite Server administrators are presented with the option of cloning a channel, as well. Refer to Section 4.7 *Cloning Software Channels* for instructions.
4. On the **New Software Channel** page, define the details of the channel following the instructions on the page. Remember, for most channel management actions, the Channel Label will be used to identify the channel, so select a meaningful label. View the details of existing channels for ideas.

   The GPG key URL will be the location of the key on the server, as defined during the client configuration process. Refer to the *Red Hat Network Client Configuration Guide*. The GPG key ID will be it's unique identifier, such as "DB42A60E", while the GPG key fingerprint will look something like "CA20 8686 2BD6 9DFC 65F6 ECC4 2191 80CD DB42 A60E".
5. When finished, click **Create Channel** at the bottom of the page.

4.6. Assigning Packages to Software Channels

When packages are uploaded initially, they can be assigned to a custom channel, multiple custom channels, or no channel at all. (See Chapter 6 *Uploading and Maintaining Custom Packages* for instructions.) Once uploaded, they may be reassigned between custom channels and the No Channels repository.

These functions are made available by clicking the **Channels** tab in the top navigation bar and then **Manage Software Channels** in the left navigation bar. In the **Software Channel Management** page, click the name of the channel to receive packages.

In the **Managed Software Channel Details** page, click the **Packages** tab and then the **Add** subtab. To associate packages with the channel being edited, select the option now containing the packages from the **View** dropdown menu and click **View**. Packages already associated with the channel being edited are not displayed. Packages not assigned to a specific channel can be found in the `Packages in no channels` menu item. Selecting `All managed packages` will present all available packages.

After clicking **View**, the package list for the selected option will appear at the bottom. Note that the page header still lists the channel being edited. In the list, select the checkboxes of the packages to be assigned to the edited channel and click **Add Packages** at the bottom-right corner of the page. A

confirmation page will appear with the packages listed. Click **Confirm** to associate the packages with the channel. The **List/Remove** subtab of the **Managed Software Channel Details** page will appear with the new packages listed.

Once packages are assigned to a channel, the Errata cache will be updated to reflect the changes. But this is delayed briefly so that users may finish editing a channel before all of the changes are made available. To initiate your changes to the cache manually, click the **here** link within the text at the top of the **List/Remove** subtab.

4.7. Cloning Software Channels

RHN Satellite Server Channel Administrators also have the ability to clone software channels for easy package association. Cloning offers you a complete replica of another channel, enabling you to immediately associate appropriate packages and Errata with a custom software channel. To access this functionality, go to the **Software Channel Management** page and click **clone channel** at the top-right corner.

You are immediately presented with three cloning options: current state of the channel, original state of the channel, or select Errata. These options are described fully on the webpage itself but can be summarized as:

- **Current state of the channel** — All of the Errata and all of the latest packages now in the target channel.
- **Original state of the channel** — All of the original packages from the target channel but none of the Errata or associated update packages.
- **Select Errata** — All of the original packages from the target channel with the ability to include certain Errata and associated update packages.

Select the option you desire using the radio buttons within the **Clone** field, identify the target channel using the **Clone From** dropdown menu, and click **Create Channel**.

On the **New Software Channel** page, complete the fields as described in Section 4.5 *Creating a Software Channel*. Note that the default values will often suffice. When satisfied, click **Create Channel**. If you selected either the original or current option, you are directed to the **Details** tab of **Managed Software Channel Details** page, where you may alter settings for the new channel. Refer to Section 4.3 *Managed Software Channel Details* for instructions.

If you used the select Errata option to clone the channel, you are instead directed to the **Clone** subtab of **Managed Software Channel Details** page, where you may individually select Errata and associated packages for cloning and inclusion in the new channel. Refer to Section 4.3 *Managed Software Channel Details* for specific instructions.

4.8. Deleting Software Channels

RHN Satellite Server and RHN Proxy Server administrators must also have the ability to remove unused channels. This action is conducted within the **Details** tab of the **Managed Software Channel Details** page. After opening up this tab, described in detail in Section 4.3 *Managed Software Channel Details*, click **delete channel** at the top-right corner of the page to remove the channel (and all packages exclusively associated with it) entirely.

Removing a channel via the website will automatically delete all packages associated with it exclusively. Packages also associated with other channels will be retained. Keep in mind, if you've established that channel on a Proxy connected to a Satellite, you will have to delete the channel on the Proxy.

Chapter 5.
Custom Errata Management

Custom Errata enable customers who have their own private channel to issue Errata Alerts for their packages. All Errata management activities will take place in the **Errata** tab of the RHN website. The instructions here should be used in conjunction with the RHN website chapter of the *RHN Management Reference Guide*.

5.1. Manage Errata

In addition to the buttons and pages available to standard RHN Management-level users, RHN Satellite Server and RHN Proxy Server customers also have access to **Manage Errata** in the left navigation bar. This button opens the **Errata Management** interface, where all custom Errata management work occurs.

Warning

Customers using both RHN Proxy Server and RHN Satellite Server should be managing Errata only on the Satellite, since the Proxies should be receiving updates directly from it. Managing Errata on a Proxy in this combined configuration risks putting your servers out-of-sync.

Clicking an Advisory within the Errata Management list takes you to the **Details** tab of the **Managed Errata Details** page. Refer to Section 5.2 *Managed Errata Details* for a full explanation of this area.

5.1.1. Published Errata

The **Published Errata** page is shown by default when you click **Manage Errata** in the left navigation bar. It displays the Errata Alerts your organization has created and disseminated.

To edit an existing published Errata, follow the steps described in Section 5.3 *Creating and Editing Errata*. To distribute the Errata, click **Send Notification** on the top-right corner of the **Errata Details** page. The Errata Alert will be sent to the administrators of all affected systems.

5.1.2. Unpublished Errata

The **Unpublished Errata** page appears when you click **Unpublished** below **Manage Errata** in the left navigation bar. It displays the Errata Alerts your organization has created but not yet disseminated.

To edit an existing unpublished Errata, follow the steps described in Section 5.3 *Creating and Editing Errata*. To publish the Errata, click **Publish Errata** on the top-right corner of the **Errata Details** page. The Errata Alert will be shifted to the **Published** page awaiting dissemination.

5.2. Managed Errata Details

If you click on the Advisory of a managed Errata Alert in the **Published** or **Unpublished** pages, its **Managed Errata Details** page appears. This page is further divided into three tabs: **Details**, **Channels**, and **Packages**.

- **Details** — Provides the primary information you entered about the custom Errata Alert during its creation. This includes a synopsis, advisory name and type, related product, bugs, description, solution, keywords, references, and notes. To change any of this information, make your modifications in the appropriate fields and click **Update Errata**.
- **Channels** — Shows the channels associated with the selected Errata. To change these associations, select or unselect the appropriate checkboxes and click the **Update Channels** button.
- **Packages** — Enables you to manage the packages associated with the selected Errata. This tab contains two subtabs that allow you to view, add, and remove packages: **List/Remove** and **Add**.
  - **List/Remove** — Displays all of the packages currently associated with the custom Errata and provides a means to cancel that association. To remove packages from the Errata, select their checkboxes and click **Remove Packages** on the bottom right-hand corner of the page. A confirmation page appears with the packages to be removed listed. Click **Confirm** to complete the action.
  - **Add** — Enables the addition of packages to the Errata. To see available packages, select an option from the **View** dropdown menu and click **View**. To add packages to the Errata you are editing, select the appropriate checkboxes and click **Add Packages**. Refer to Section 5.4 *Assigning Packages to Errata* for a comprehensive discussion of this process.

5.3. Creating and Editing Errata

To make a custom Errata Alert, click the **Manage Errata** button in the **Errata** category of the RHN website. Click the **create new erratum** button at the top-right corner of the **Errata Management** page.

Enter an intuitive label for the Erratum in the **Advisory** field, ideally following a naming convention adopted by your organization. Note that this label cannot begin with the letters "RH" (capitalized or not) to prevent confusion between custom Errata and those issued by Red Hat.

Then, complete all remaining required fields and click the **Create Errata** button. View standard Red Hat Errata Alerts for examples of properly completed fields.

RHN Satellite Server administrators may also create Errata by cloning an existing one. This cloning preserves package associations and simplifies issuing Errata. Refer to Section 5.5 *Cloning Errata* for instructions.

To edit an existing Errata Alert's details, click its Advisory in the **Errata Management** page, make your changes in the appropriate fields of the **Details** tab, and click the **Update Errata** button. Click on the **Channels** tab to alter the Errata's channel association. Click on the **Packages** tab to view and modify its packages.

To delete Errata, select their checkboxes in the **Errata Management** page, click the **Delete Errata** button, and confirm the action. Note that deleting published Errata may take a few minutes.

Tip

If you want to receive an email when Errata Alerts are issued for your systems, go to **Your RHN** => **Your Preferences** in the RHN Website and select **Receive email notifications**.

5.4. Assigning Packages to Errata

In the **Managed Errata Details** page, click the **Packages** tab and then the **Add** subtab. To associate packages with the Errata being edited, select the option now containing the packages from the **View** dropdown menu and click **View**. Packages already associated with the Errata being edited are not displayed. Selecting `All managed packages` will present all available packages.

After clicking **View**, the package list for the selected option will appear at the bottom. Note that the page header still lists the Errata being edited. In the list, select the checkboxes of the packages to be assigned to the edited Errata, and click **Add Packages** at the bottom-right corner of the page. A confirmation page will appear with the packages listed. Click **Confirm** to associate the packages with the Errata. The **List/Remove** subtab of the **Managed Errata Details** page will appear with the new packages listed.

Once packages are assigned to an Errata, the Errata cache will be updated to reflect the changes. But this is delayed briefly so that users may finish editing an Errata before all of the changes are made available. To initiate your changes to the cache manually, click the **here** link within the text at the top of the **List/Remove** subtab.

5.5. Cloning Errata

RHN Satellite Server customers also have the ability to clone Errata for easy replication and distribution. Only Errata potentially applicable to one of your channels can be cloned. An Errata is potentially applicable to a channel if that channel was cloned from a channel to which the Errata applies. To access this functionality, click **Errata** in the top navigation bar, then **Clone Errata** in the left navigation bar. This button will appear only for RHN Satellite Server customers.

Once in the **Clone Errata** page, select the channel containing the Errata from the **View** dropdown menu and click **View**. Once the Errata list appears, select the checkbox of the Errata to be cloned and click **Clone Errata**.A confirmation page will appear with the Errata listed. Click **Confirm** to finish the cloning.

The cloned Errata will appear in your unpublished errata list. From there you can verify the Errata text and the packages associated with that Errata. Once you are ready, you can publish the Errata so it is available to users in your organization.

Chapter 6.

Uploading and Maintaining Custom Packages

Depending upon which Red Hat Network service is used, there are two different mechanisms for uploading packages to private channels.

Customers of the RHN Proxy Server will use the **RHN Package Manager** application, which sends package header information to the central Red Hat Network Servers and places the package itself into the local repository of the RHN Proxy Server from which **RHN Package Manager** is invoked.

Customers of the RHN Satellite Server will use the **RHN Push** application, which sends package header information to the local RHN Satellite Server and places the package into the local repository of the RHN Satellite Server from which **RHN Push** is invoked.

This chapter discusses both of these tools in detail.

Warning

Remember, customers using both RHN Proxy Server and RHN Satellite Server should be using only the **RHN Push** because this combination requires custom packages and channels be uploaded to the Satellite only. From there, the Proxies will obtain the packages and distribute them to client systems.

6.1. Uploading Packages to RHN Proxy Server

The **RHN Package Manager** allows an organization to serve custom packages associated with a private RHN channel through the RHN Proxy Server. If you want the RHN Proxy Server to update only official Red Hat Enterprise Linux packages, you do not need to install the **RHN Package Manager**.

To use the **RHN Package Manager**, install the `rhns-proxy-package-manager` RPM package and its dependencies. This package is available to registered RHN Proxy Server systems and may be installed by running `up2date rhns-proxy-package-manager`.

Only the header information for the packages is uploaded to the RHN Servers. The headers are required so that RHN can resolve package dependencies for the client systems. The actual package files (`*.rpm`) are stored on the RHN Proxy Server. For this reason, custom packages cannot be downloaded through the RHN website, although they are listed. They must be retrieved by the client system using `up2date`.

6.1.1. Configuring and Using the RHN Package Manager

Note

It is recommended that at least one private channel be created to receive custom packages prior to upload, since a channel is required for systems to obtain the packages.

The following command uploads the package headers to the RHN Servers and copies the packages to the RHN Proxy Broker Server:

```
rhn_package_manager -c label_of_private_channel pkg-list
```

The *label_of_private_channel* is the custom channel created to receive these packages. Be sure you use the precise channel label specified during its creation. If you have one or more channels specified (using -c or --channel), the uploaded package headers will be linked to all the channels identified. If you do not specify a channel, the packages will be deposited in the **No Channels** section of the **Package Management** page. Refer to Section 4.6 *Assigning Packages to Software Channels* for instructions on reassigning packages.

The *pkg-list* reference represents the list of packages to be uploaded. Alternatively, use the -d option to specify the local directory that contains the packages to be added to the channel. **RHN Package Manager** can also read the list of packages from standard input (using --stdin).

Other options, such as the Red Hat Network Server URL, the HTTP proxy username and password (if your HTTP proxy requires authentication), and the top directory where packages will live, can be specified in a configuration file. **RHN Package Manager** will try to read the configuration file specified using the -f or --conffile options, if present.

Parameters not set in this file will be read from .rhn_package_manager in the home directory of the user currently logged in and finally from /etc/rhn/rhn_package_manager.conf. Make sure all of these files have the appropriate permissions to prevent others from reading them.

After uploading the packages, you can check to see if the local directory is in sync with the RHN Server's image of the channels:

```
rhn_package_manager -s -c name_of_private_channel
```

This -s option will list all the missing packages (packages uploaded to the RHN Server but not present in the local directory). You must be an Organization Administrator to use this option. The script will prompt you for your RHN username and password.

You can also use the **RHN Package Manager** to retrieve a list of packages in a channel (as stored by the RHN Server):

```
rhn_package_manager -l -c name_of_private_channel
```

The -l option will list the package name, version number, release number, architecture, and channel name for each package in the specified channel(s). Refer to Table 6-1 for additional options.

Here is a summary of all the command line options for **RHN Package Manager** (rhn_package_manager):

Option	Description
-v, --verbose	Increase verbosity
-d, --dir *DIRECTORY_NAME*	Process packages from this directory
-f, --conffile *CONFFILE_NAME*	Configuration file. If a configuration file is not given, ~/.rhn_package_manager is used. If ~/.rhn_package_manager is not found, /etc/rhn/rhn_package_manager.conf is used.
-c, --channel *CHANNEL_NAME*	Specify the channel to receive packages. Multiple channels may be specified using multiple instances of -c (e.g.: -c channel_one -c channel_two)
-e, --exclude *PACKAGE_NAME*	Filter out packages by name.
-n, --count *NUMBER*	Process this number of headers per call — the default is 32.
-l, --list	List the packages in the specified channel(s).

Option	Description
`-s, --sync`	Check if local directory is in sync with the server.
`-p, --printconf`	Print the current configuration and exit.
`--newest`	Only push the packages that are newer than packages already pushed to the server for the specified channel.
`--nosig`	Don't fail if packages are unsigned.
`--no-ssl`	Turn off SSL (not recommended).
`--stdin`	Read the package names from stdin.
`--username` *USERNAME*	Specify RHN username. If not provided, you will be prompted for the username of a valid Channel Administrator.
`--password`	Specify RHN password. If not provided, you will be prompted for the password of a valid Channel Administrator.
`--source`	Upload the listed packages as source packages instead of binary packages.
`--dontcopy`	In the post-upload step, do not copy the packages to their final location in the package tree.
`--test`	Only print the packages to be pushed.
`-?, --help`	Display the help screen with a list of options.
`--usage`	Briefly describe the available options.

Table 6-1. `rhn_package_manager` options

Tip

These command line options are also described in the `rhn_package_manager` man page: `man rhn_package_manager`.

6.2. Uploading Packages to RHN Satellite Server

The **RHN Push** application allows an organization to serve custom packages associated with a private RHN channel through the RHN Satellite Server. If you want the RHN Satellite Server to update only official Red Hat Enterprise Linux packages, you do not need to install the **RHN Push** application.

To use the **RHN Push** application, install the `rhnpush` package and its dependencies. This package is available to registered RHN Satellite Server systems and may be installed by running `up2date rhnpush`.

The **RHN Push** application uploads RPM header information to the RHN Satellite Server database and places the RPM in the RHN Satellite Server package repository. Unlike the RHN Proxy Server's **RHN Package Manager**, **RHN Push** never distributes package information, even the headers, beyond the RHN Satellite Server database.

6.2.1. Configuring and Using the RHN Push application

Note

It is recommended that at least one private channel be created to receive custom packages prior to upload, since a channel is required for systems to obtain the packages.

The following command uploads package headers to the RHN Satellite Server and copies the packages to the RHN Satellite Server package repository:

```
rhnpush -c label_of_private_channel --server localhost pkg-list
```

The *label_of_private_channel* is the custom channel created to receive these packages. Be sure you use the precise channel label specified during its creation. If you have one or more channels specified (using `-c` or `--channel`), the uploaded package headers will be linked to all the channels identified. If you do not specify a channel, the packages will be deposited in the **No Channels** section of the **Package Management** page. Refer to Section 4.6 *Assigning Packages to Software Channels* for instructions on reassigning packages.

The `--server` option specifies the server to which the packages will be installed, and is required. **RHN Push** may be installed on external systems, but running **RHN Push** locally on the RHN Satellite Server is recommended.

The *pkg-list* reference represents the list of packages to be uploaded. Alternatively, use the `-d` option to specify the local directory that contains the packages to be added to the channel. **RHN Push** can also read the list of packages from standard input (using `--stdin`).

The following summary contains all command line options for the `rhnpush` command:

Option	Description
`-v --verbose`	Increase verbosity (can use multiple times).
`-d=, --dir=`*DIRECTORY*	Process packages from this directory.
`-c=, --channel=`*CHANNEL_LABEL*	Specify the channel to receive packages. Note that this is required and is not the same as the channel's name. Multiple channels may be specified using multiple instances of -c (e.g. -c=CHANNEL_ONE -c=CHANNEL_TWO).
`-n=, --count=`*N_HEADERS_PER_CALL*	Process this number of headers per call. Must be an integer. (25 is default.)
`-l, --list`	List only the specified channels.
`-r=, --reldir=`*RELATIVE_DIRECTORY*	Associate this relative directory with each file.
`-o=, --orgid=`*ORGANIZATION_ID*	Include your organization's ID number. Must be an integer.
`-u=, --username=`*USERNAME*	Include RHN username of user that has administrative access to the specified channel. If not provided, `rhnpush` will prompt for the username of a valid Channel Administrator.

Option	Description
`-p=, --password=`*PASSWORD*	Include RHN password of user that has administrative access to the specified channel. If not provided, `rhnpush` will prompt for the password of a valid Channel Administrator.
`-s, --stdin`	Read package list from standard input (a piped `ls` command, for example).
`-X=, --exclude=`*GLOB*	Exclude packages that matches this glob expression.
`--force`	Force upload of a package, even if a package of that name and version currently exists in the channel. Without this option, uploading a pre-existing package will return an error.
`--nosig`	Don't fail if packages are unsigned.
`--newest`	Push only the packages that are newer than those on the server. Please note that source RPMs are special in that their versions are never compared. Therefore, using this option with RHN Push will upload source RPMs for the specified channels only if they were not previously uploaded, and binaries built from them exist in those channels. In other words, you have to upload a binary RPM before being able to use *--newest* with its source RPM.
`--header`	Upload only the headers.
`--source`	Upload the indicated source packages. (We treat them differently.)
`--server`*SERVER*	Specify the server to which packages will be uploaded. Currently, a value of http://localhost/APP is necessary. This parameter is required.
`--test`	Only print the packages to be pushed, don't actually push them.
`-h, --help`	Briefly describe the options.
`-?, --usage`	View the usage summary.

Table 6-2. `rhnpush` options

Tip

These command line options are also described in the `rhnpush` man page: `man rhnpush`.

Index